Computer Aided Design of Digital Filters

S. Hossein Mousavinezhad
Professor and Chair
Department of Electrical and Computer Engineering
Western Michigan University
Kalamazoo, Michigan 49008
(616) 387-4057
FAX (616) 387-4024
h.mousavinezhad@wmich.edu

I. INTRODUCTION

Digital Signal Processing (DSP) is an important and growing subject area in electrical and computer engineering (ECE) with applications in many engineering/science disciplines including communication systems, biomedical engineering, speech processing, avionics, and automation. In many engineering schools ECE departments offer DSP courses taught at undergraduate and graduate levels. At Western Michigan University, we offer two courses for senior and first year graduate students covering topics such as discrete-time signals and systems, various transforms in signal processing, difference equations, digital filter design techniques/structures, power spectrum estimation, multi-dimensional DSP, finite word length effects, fast algorithms, and adaptive filters.

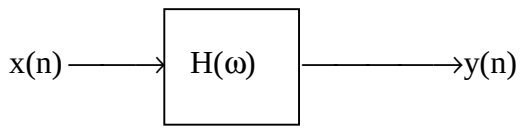
One important topic in DSP involves designing recursive and non-recursive digital filters and their implementation. Because of the complexity of analog/digital filter design algorithms and their mathematical formulations, students usually have difficulty understanding and appreciating the importance of various types of filters. The author has successfully incorporated the use of computer tools such as MATHCAD and MATLAB into the DSP courses and has implemented digital filters in real time utilizing digital hardware and filter design packages. In this paper we will present various algorithms and numerical examples for designing finite impulse response (FIR) and infinite impulse response (IIR) digital filters using commercial software packages and their real time implementation using Texas Instruments TMS320C31 DSP boards. Signal Processing Toolbox of MATLAB and the ability to simulate mathematical expressions naturally in MATHCAD are very attractive for DSP applications. It is noted that the so-called powerful tools that are available with such courses should be used along with careful classroom presentation of difficult and abstract concepts and underlying principles.

In Section II we present some theoretical background for filter design and discuss some mathematical preliminaries usually needed in a DSP course. Section III includes MATLAB and MATHCAD examples used in a senior-level course offered at our school

and a project using the ELF DSP system. Section IV covers conclusions and recommendations for further studies in this area.

II. FILTER DESIGN THEORY

Digital filters find many applications in digital communication systems, computer music, electronics and biomedical signal processing, to just mention a few. Basically a digital filter is defined by its difference equation (time domain representation) or a transfer function (frequency domain representation.) Symbolically we represent digital filters by:



where x(n) denotes the digital input and y(n) is the output at discrete (integer) time index n. H(ω) is the frequency response (transfer) function which is also the discrete-time Fourier Transform of the impulse response (IR), h(n).

The design problem is usually stated in terms of a given set of specifications in the frequency domain and the objective is to find h(n) or H(ω) that will satisfy and (in most cases) exceed the minimum requirements given in the specifications statement.

There are two important types of digital filters used in practice: Finite Impulse Response (FIR), or non-recursive or feedforward filters, and Infinite Impulse Response (IIR), or recursive or feedback filters. In literature they are also known as MA (moving average) and ARMA (autoregressive MA) filters, among others. An FIR filter is described by the following difference equation:

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + \ldots\ldots b_{N-1}x(n-N+1) =$$

$$\sum_{i=0}^{N-1} b_i x(n-i).$$

One important design method in FIR filters is the so-called windowing technique which utilizes the Fourier series theory to express the impulse response in terms of the given ideal (desired) frequency response characteristics:

$$h_d(n) = (1/2\pi)\int_0^{2\pi} H_d(\omega)\exp(j\omega n)d\omega$$

$$h(n) = h_d(n)w(n) \quad ; \quad -M \le n \le M$$

where $H_d(\omega)$ denotes the desired frequency response function and w(n) is the symmetric window of length 2M+1. Once h(n) is found one can obtain the transfer function of the filter, H(z). Substituting z = exp(jω) we can obtain the frequency response of the non-causal filter given by H(ω). Finally the causal filter is designed using the relation $H_c(\omega) = H(\omega)\exp(-jM\omega)$. Note that $|H_c(\omega)| = |H(\omega)|$. In addition to windowing/truncation

method, other methods are available for designing these filters [1], including least-squares design and equiripple design [2]. Algorithms such as Parks-McClellan are available for computing filter coefficients directly. Commercial design tools such as Digital Filter Design Package (DFDP) by the Atlanta Signal Processors, Inc. (ASPI) can also be used to obtain filter coefficients for a given set of specifications. Then the filter can be implemented, in real-time, using a variety of available digital signal processors, e.g., Texas Instruments TMS320C31 DSP board which is also available through ASPI's ELF System. In the next section we present one numerical example using Fourier series method and simulation using MATHCAD. It is noted that although FIR design methods are straightforward (as compared to IIR design) and they can be designed to have linear phase, the order of the filter (2M+1) is usually large for typical design problems. Another advantage of these filters is that they can be used to design multiband filters, Hilbert transformers, differentiators, in addition to the usual (LP, HP, BP, BR) filters.

IIR filters are usually designed starting with analog filters and applying appropriate transformations such as impulse invariance or bilinear transform. Therefore the starting point can be the transfer function of the analog filter, H(s), for which excellent design methods and appropriate approximations such as Butterworth and Chebychev methods are available. From H(s) and applying bilinear transformation $s = 2(z-1)/(T(z+1))$, we obtain the transfer function H(z) of the digital filter ($T=1/F_s$ is the sampling interval or period with F_s being the sampling frequency or rate.) These filters usually require lower order implementation compared to FIR filters, but their phase responses, in general, are non-linear.

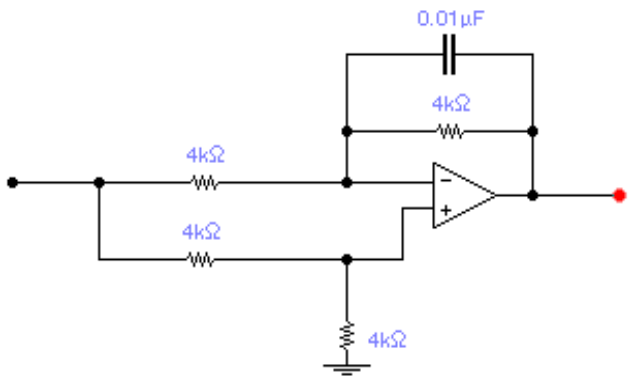
The difference equation of a recursive filter with x(n) as input and y(n) as output or response is given as:

$$y(n) + a_1y(n-1) + a_2y(n-2) + a_3y(n-3) + \ldots\ a_Ny(n-N) =$$
$$b_0x(n) + b_1x(n-1) + b_2x(n-2) + \ldots\ b_Mx(n-M).$$

In the next section we present examples of FIR and IIR filter simulations using MATLAB and MATHCAD. More examples will be provided during the 1998 ASEE Annual Conference in Seattle, Washington.

III. COMPUTER EXAMPLES

As a first example, consider the active filter designed using RC + OP-AMP circuit shown bellow. For the purpose of filter problems we assume that the OP-AMP is ideal and operating in its linear region. More complicated filters can be designed using several stages in cascade resulting in filters with higher orders. For the purpose of illustration we have chosen a first-order filter.



Applying circuit laws one obtains the expression for the transfer function as:

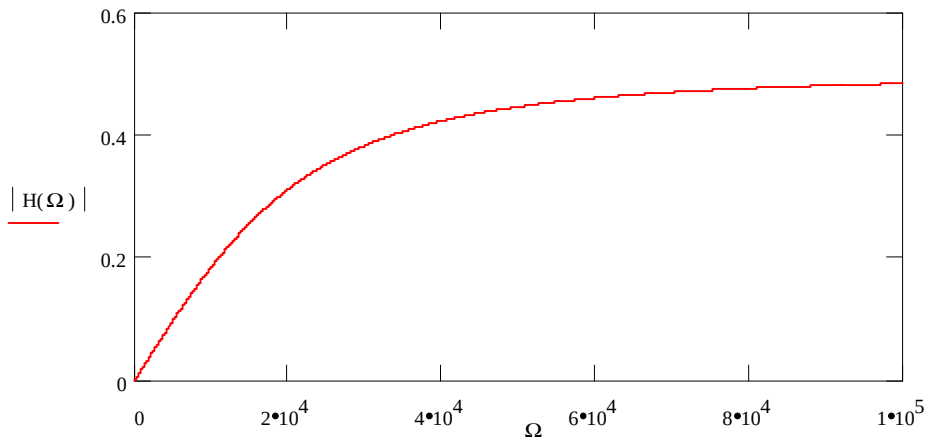
$$H(s) = V_{out}(s)/V_{in}(s) = \frac{s}{2s + 50{,}000}$$

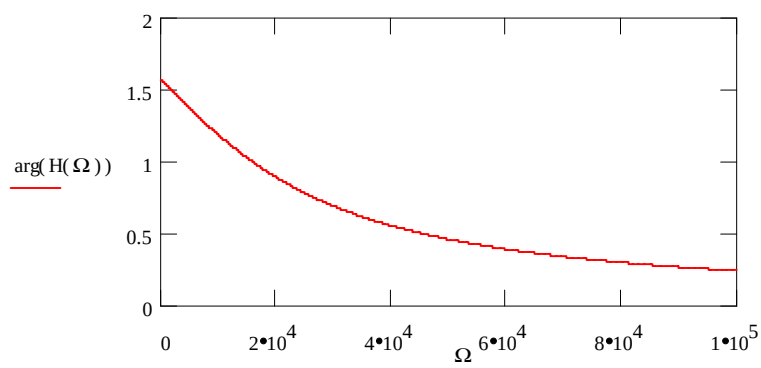
We now use MATHCAD to obtain magnitude and phase plots of this transfer function:

$j := \sqrt{-1}$

$\Omega := 0, 1 .. 100000$

$$H(\Omega) := j \cdot \frac{\Omega}{(2 \cdot j \cdot \Omega + 50000)}$$



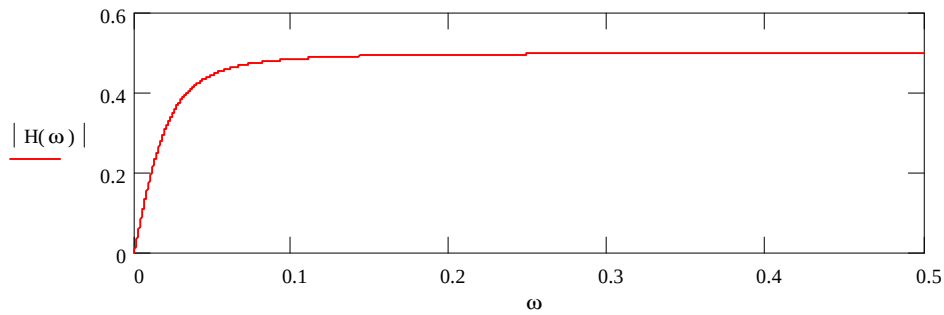


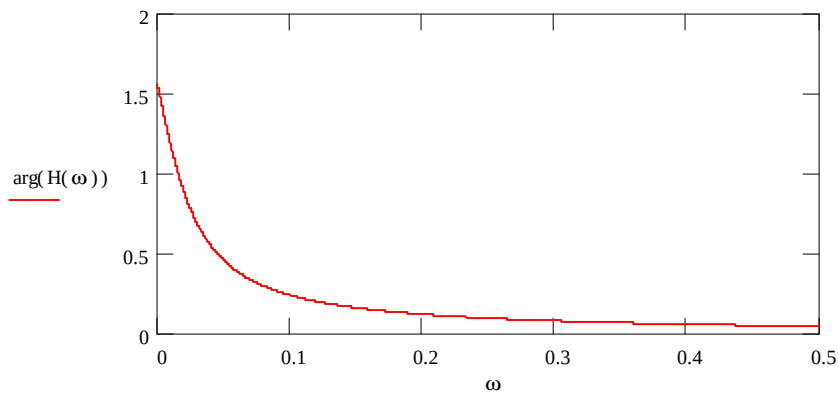
Using the bilinear transformation s = 2(z-1)/[T(z+1)] with T = 1 μsec, we transform H(s) to get H(z) of the equivalent digital filter. Finally substituting z = exp(jω) we get the frequency response function H(ω) and plot magnitude and phase responses as a function of digital frequency, ω. A MATHCAD simulation is shown bellow. Note that we proper choice of sampling interval, this method results in a good digital filter equivalent of the continuous-time (analog) filter. The bilinear transformation is widely used and with MATLAB (+ SP Toolbox's, "bilinear" function) one can also design a variety of analog filters employing Chebyshev, Butterworth and Elliptic filter approximations.

$j := \sqrt{-1}$

$\omega := 0, 0.0001 .. 0.5$

$$H(\omega) := 2\cdot 10^6 \cdot \frac{(\exp(j\cdot\omega) - 1)}{\left[\left(4\cdot 10^6 + 50000\right)\cdot \exp(j\cdot\omega) - 4\cdot 10^6 + 50000\right]}$$





Finally, we consider an FIR design problem where a (2M+1)-tap linear phase digital filter is designed to approximate the given desired (ideal) multi-band filter:

$H_d(\omega) = 1$, for $-\pi/4 \le \omega \le \pi/4$ and $H_d(\omega) = 2$, for $-\pi/2 \le \omega \le -\pi/4$ & $\pi/4 \le \omega \le \pi/2$;
$H_d(\omega) = 0$, otherwise in the rest of the interval.

We will use symmetric Hanning window:

$w(n) = 0.5 + 0.5\cos(\pi n/M)$ for $-M \le \omega \le M$, $w(n) = 0$, otherwise.

$h_d(n) = Sa(n\pi/2) - Sa(n\pi/4)$ with Sa being the sampling function and $h_d(n=0) = 0.75$.

With this information one find the frequency response function as:

$$H(\omega) = 0.75 + 2\sum_{n=1}^{M} hd(n)w(n)\cos(n\omega)$$

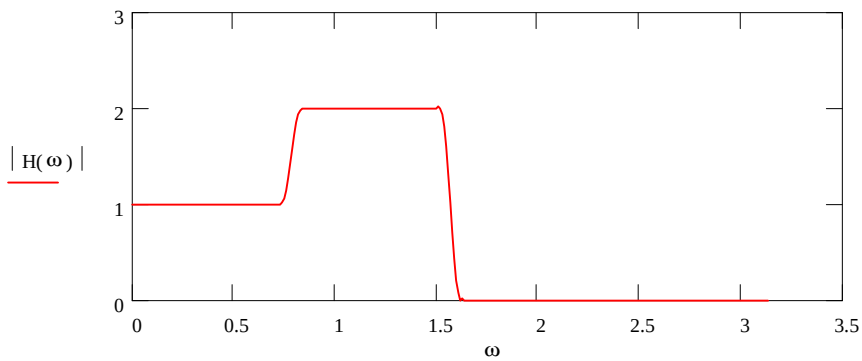
We now show a MATHCAD simulation:

$M := 101 \qquad \omega := 0, 0.01 .. \pi \qquad n := 1, 2 .. M$

$$hd(n) := 2\cdot\left[\frac{\sin\left(n\cdot\frac{\pi}{2}\right)}{(\pi\cdot n)}\right] - \frac{\sin\left(n\cdot\frac{\pi}{4}\right)}{(\pi\cdot n)}$$

$$w(n) := 0.5 + 0.5\cdot\cos\left(\pi\cdot\frac{n}{M}\right)$$

$$H(\omega) := 0.75 + 2\cdot\sum_{n} hd(n)\cdot w(n)\cdot\cos(n\cdot\omega)$$



IV. CONCLUSIONS

Digital signal processing is an important and growing subject area in electrical and computer engineering field and many schools offer courses in the undergraduate as well as graduate levels. The mathematical preliminaries are such that to benefit from this subject, students need good grounding in such topics as Laplace transforms and solution of differential equations, frequency response, Fourier methods and some experience in dealing with probability functions and random signals. Of course it is possible to cover "Fundamentals of DSP" in a signals & systems course or, as in [3], in an introductory engineering course (perhaps for ECE and other engineering majors.) But it has been the author's experience after nearly ten years of graduate/undergraduate DSP instruction that even senior/first-year graduate students still find some of the topics too advanced and "highly mathematical". At the end one will realize that there is a proper place and time for offering such a course, and more important, introduce students to power and limitations of commercial software packages. Our students have found both MATLAB and MATHCAD useful and with availability of student editions and the fact that, for

example in mathcad you can use a “matlab component”, these tools can be used in a complementary manner.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Leonard Lamberson, Dean of Engineering at Western Michigan University for his support and encouragement, Mr. Sridhar Erra, Director of Computer-Aided Engineering Center, and Mr. David Florida for computer hardware/software support. I would also like to express my thanks to MathSoft Inc., Mr. Tom Robbins of Prentice Hall College Publications, and Mr. Joseph Koenig of Interactive Image Technologies, for their support.

REFERENCES

[1]. Proakis, J. G. and Manolakis, D. G., “Introduction to Digital Signal Processing, Principles, Algorithms, and Applications,” third edition, Prentice Hall, 1996.

[2]. Porat, B., “A Course in Digital Signal Processing,” John Wiley, 1997.

[3]. McClellan, J. H., Schafer, R. W. Schafer, and Yoder, M.A., “DSP First, A Multimedia Approach,” Prentice Hall, 1998.

Biographical Information

S. HOSSEIN MOUSAVINEZHAD

Dr. Mousavinezhad is Professor and Chair, Department of Electrical and Computer Engineering, Western Michigan University, Kalamazoo, Michigan. His expertise focuses on the applications of digital signal processing in communication systems and bioelectromagnetics. He is currently conducting research into the development of signal processing algorithms and in the area of global aspects of engineering education.